Dados e Aprendizagem Automática
Report - Grupo MEI 8

Abhimanyu Aryan - PG51632

Bernard Georges - PG53698

José Fonte - a91775

João Braga - PG53951

1 Introduction

The primary objective of this report is to elucidate the methods and decisions undertaken throughout the execution of the project. It aims to provide a comprehensive understanding of the strategies employed and choices made during the completion of both the group-specific dataset exploration and the Kaggle competition with the 2023/24 DAA class.

2 Task One - Group Dataset - Spotify Tracks

For our initial task, the team opted to investigate a publicly available dataset from the Hugging Face Community. This dataset provides comprehensive details about Spotify tracks, encompassing various music-oriented attributes. The richness of this dataset offers diverse avenues for exploration and multiple potential objectives.

In this project, our focus was on predicting the popularity of music based on its characteristics, such as genre, loudness, and other relevant factors. By leveraging this dataset, we aimed to uncover patterns that correlate with the popularity of tracks, contributing to a deeper understanding of music preferences.

For those interested in this domain, the dataset accessible through the following **link**.

2.1 Work Methodology

The SEMMA methodology— **Sample, Explore, Modify, Model, and Assess**—is a widely used approach in data science. In the interest of promoting an accurate and actionable outcome our group chose the mentioned method due to its systematic, effective, structured and rigorous analytical process.

We start by selecting a representative sample, move on to exploring the data for patterns, and then modify it for quality. The core is in the Model phase, where predictive models are developed, and finally, the Assess phase evaluates the model's performance.

2.2 Tools and Libraries Used

- JupyterLab : A development environment that facilitates creating and sharing documents containing live code, equations, visualizations, and narrative text.
- Pandas : A powerful data manipulation library in Python that provides data structures like DataFrames, making it efficient to analyze and manipulate structured data.
- NumPy : NumPy is the fundamental package for scientific computing in Python. It provides support for large, multi-dimensional arrays and matrices, along with mathematical functions to operate on these elements.
- Matplotlib : A widely used plotting library in Python, Matplotlib enables the creation of static, animated, and interactive visualizations in Python. It's versatile and essential for data exploration and presentation.
- Seaborn : Seaborn is a statistical data visualization library that works in conjunction with Matplotlib. It simplifies the process of creating aesthetically pleasing and informative statistical graphics.
- Sklearn : Scikit-learn is a machine learning library for Python. It features various tools for classification, regression, clustering, and more.

2.3 Sample - Dataset Overview

Although the group did not follow this phase to its core which is to gather data, it is still important to understand the domain of the chosen dataset. For this the group needs to comprehend each column of the data, the main help for this phase are the descriptions written by the maintainers of hugging face community which allows the group to fully grasp the meaning of each feature.

- **track_id**: The Spotify ID for the track
- **artists**: The artists' names who performed the track. If there is more than one artist, they are separated by a ;
- **album_name**: The album name in which the track appears
- **track_name**: Name of the track
- **popularity[target]**: The popularity of a track is a value between 0 and 100, with 100 being the most popular. The popularity is calculated by algorithm and is based, in the most part, on the total number of plays the track has had and how recent those plays are. Generally speaking, songs that are being played a lot now will have a higher popularity than songs that were played a lot in the past. Duplicate tracks (e.g. the same track from a single and an album) are rated independently. Artist and album popularity is derived mathematically from track popularity.
- **duration_ms**: The track length in milliseconds
- **explicit**: Whether or not the track has explicit lyrics (true = yes it does; false = no it does not OR unknown)
- **danceability**: Danceability describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. A value of 0.0 is least danceable and 1.0 is most danceable
- **energy**: Energy is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy. For example, death metal has high energy, while a Bach prelude scores low on the scale
- **key**: The key the track is in. Integers map to pitches using standard Pitch Class notation. E.g. 0 = C, 1 = C/D, 2 = D, and so on. If no key was detected, the value is -1
- **loudness**: The overall loudness of a track in decibels (dB)
- **mode**: Mode indicates the modality (major or minor) of a track, the type of scale from which its melodic content is derived. Major is represented by 1 and minor is 0
- **speechiness**: Speechiness detects the presence of spoken words in a track. The more exclusively speech-like the recording (e.g. talk show, audio book, poetry), the closer to 1.0 the attribute value. Values above 0.66 describe tracks that are probably made entirely of spoken words. Values between 0.33 and 0.66 describe tracks that may contain both music and speech, either in sections or layered, including such cases as rap music. Values below 0.33 most likely represent music and other non-speech-like tracks
- **acousticness**: A confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence the track is acoustic

- **instrumentalness**: Predicts whether a track contains no vocals. "Ooh" and "aah" sounds are treated as instrumental in this context. Rap or spoken word tracks are clearly "vocal". The closer the instrumentalness value is to 1.0, the greater likelihood the track contains no vocal content
- **liveness**: Detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live. A value above 0.8 provides strong likelihood that the track is live
- **valence**: A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry)
- **tempo**: The overall estimated tempo of a track in beats per minute (BPM). In musical terminology, tempo is the speed or pace of a given piece and derives directly from the average beat duration
- **time_signature**: An estimated time signature. The time signature (meter) is a notational convention to specify how many beats are in each bar (or measure). The time signature ranges from 3 to 7 indicating time signatures of 3/4, to 7/4.
- **track_genre**: The genre in which the track belongs

GOAL : Predict Tracks Popularity based on its features

2.4 Explore - Dataset Exploration and Visualization

In this phase the group explores the dataset, extracting knowledge through some statistics and plots. This phase gave the group a deeper view of the features and the relation between them, as well as an understanding of some of its problems (such as missing values, repeated tracks, etc.). The following points are some of the takeaways the group had from this analysis:

- **Matrix Size**
  - 114000 entries with 21 features
- **Features Type**
  - a lot of diferent types : string, bool, float64 and int64
- **Missing Values**
  - one missing value in 3 features
- **Unique Values**
  - number of track_ids different from number of ids, so there's repeated tracks
  - one thousand tracks per genre
- **Quick Stats of Number Features (count, mean, std and 4 quartiles**)
  - duration is in the millisecond scale (very high values)
  - 75% of duration values are under 26150 ms
  - loudness is a negative value (due to decibels representation)
  - 75% of instrumentalness values are below 0.049 (very low values)
- **Histograms/Bar Chart of Features**

  - popularity - a lot of 0s and very few values above 85, as expected.
  - duration_ms - normal dist with positive skewness and positive kurtosis | a few outliers grow the scale in one unit.
  - explicit - binary feature not balanced.
  - danceability - normal dist with negative skewness and negative kurtosis
  - energy - linear dist with positive decline. negative skewness and kurtosis
  - loudness - normal dist with negative skewness and positive kurtosis
  - mode - unbalanced feature, one value is almost the double of other.
  - speechiness - normal dist with positive skewness and positive kurtosis | 75% of values below 0.084500
  - instrumentalness - very low values, almost near zero.
  - valence - normal dist with negative kurtosis
  - tempo - most tempos between 100-150bpm. some entries are 0 bpm, which doesn't make sense.
  - time_signature - most of tracks on 4/7

- **Correlation Analysis**
  - High correlation between some features
    * loudness-energy(+0.76)
    * acousticness-energy(-0.73)
    * acousticness-loudness(-0.59)
    * danceability-valence(+0.48)
    * loudness-instrumentalness(-0.43)
    * valence-instrumentalness(-0.32)
    * explicit-speechiness(0.31)
  - PairPlot between numeric features
  - Deeper analysis between features
  - Highly Correlated features and the target feature

- **All Types of Track Genre**
  - lot of sub-genres like "latin" and "latino" or "electro" and "electronic".

2.5 Modify - Basic Data Preparation

In this phase the data undergoes a basic preparation so we can test a baseline model. This phase follows the given structure:

- Dropping Features.

  Rationale : These features do not contribute as a valid metric and shouldn't integrate the modelling phase.
  - Drop id
  - Drop track_id
  - Drop artist
  - Drop track_name
  - Drop album_name

- Dealing with column types
  - explicit (boolean -> binary)
  - track_genre (one hot encoding)
- Dealing with Missing Values

2.6 Model - Baseline Model

The group followed the professor's advice and initially modeled the problem with minimal to no data preparation, allowing for a better assessment of the influence of data preparation on the models performance. Given the nature of the target variable as a continuous value, the following Regression Models were employed:

- **Regression Tree** - Score: 0.028326331711703734
- **Linear Regression** - MSE: 373.9630315451624 | MAE: 14.181756090937306 | RMSE: 19.338123785547612

2.7 Modify - Advanced Data Preparation

In this section, we advance our data preparation techniques beyond the basics, aiming to optimize the dataset for improved model performance. This phase follows the structure:

- Some of the same Basic Preparation procedures
  - Dropping features - same as previous preparation (except 'track_id')
  - Dealing with column types (only 'explicit')
- Map Music Sub-genres to Main-Genre
- One Hot Encoding 'main_genre'
- Aggregation of Repeated Tracks based on 'track_id'
  - Aggregation of Repeated Tracks
  - Dropping 'track_id' feature
- Outliers Treatment
- Feature Scaling (Normalization)
- Feature Selection
  - SelectKBest Model
  - RandomForrestRegressor Model
  - Remembering the Correlation Analysis
  - Dropping Least Important Features Features

2.8 Model - Models with Advanced Data Preparation

- Models with advanced preparation without Feature Selection
  - **Decision Tree** - R-squared: 0.12793 | MSE: 0.038952 | MAE: 0.16108 | RMSE: 0.19736
  - Linear Regression - R-squared: 0.071901 | MSE: 0.03918 | MAE: 0.16175 | RMSE: 0.19796
  - **Support Vector Regression** - R-squared: 0.23408 | MSE: 0.032340 | MAE: 0.13721 | RMSE: 0.17983
- Models with advanced preparation with Feature Selection
  - **Decision Tree** - R-squared: 0.13159 | MSE: 0.037105 | MAE: 0.14946 | RMSE: 0.19262
  - **Linear Regression** - R-squared : 0.072185 | MSE: 0.039242 | MAE: 0.16181 | RMSE: 0.19809
  - **Support Vector Regression** - R-squared: 0.22929 | MSE: 0.032542 | MAE: 0.13747 | RMSE: 0.18039
  - **Random Forrest Regressor** - R-squared: 0.32047 | MSE: 0.028773
  - **Cross-Validation** - Best Score: 0.33402185

2.9 Assessement

Despite our best efforts, the assessment of the results reveals that they did not align with our initial expectations. The intricacies of the dataset posed challenges in extracting meaningful insights and knowledge. Despite the difficulties, it's important to acknowledge that the process of navigating through a complex dataset itself is valuable. The experience gained in handling and attempting to extract knowledge from a challenging dataset contributes to our understanding of the intricacies involved in real-world data analysis. While the results may not have met our initial expectations, the journey itself has provided valuable insights and learning opportunities.

3 Task Two - Kaggle Competition

For the second task of the assignment the group was challenged to participate in a private Kaggle competition among the MSc students. The competition revolves around the pivotal task of accurately predicting the electrical energy generated by solar panels and injected into existing electrical grids at each hour of the day. As the world increasingly turns to solar energy as a key player in the transition to clean and renewable energy sources, this competition gains significance in addressing the practical challenges associated with optimizing solar energy use and harmoniously integrating solar systems into electricity grids.

In this report, we explore the objectives, dataset, evaluation criteria, and submission requirements of the competition. The focus is on understanding the intricacies of energy forecasting, leveraging machine learning techniques to enhance accuracy, and contributing to the broader goals of energy efficiency, reduced greenhouse gas emissions, and overall environmental sustainability.

3.1 Dataset Description

The competition provides two types of datasets: one containing energy data and another containing weather data. Both types of datasets are divided by date, with one covering the period from September 2021 to December 2021 and the other covering the entire year of 2022. Four datasets are available for training and tuning Machine Learning models:

- energia_202109-202112.csv
- energia_202201-202212.csv
- meteo_202109-202112.csv
- meteo_202201-202212.csv

These datasets include records for each hour of a given day and should be used for the development and training of Machine Learning models. The energy production dataset provides information on the amount of energy injected into the electricity grid for each record, labeled as the attribute "Grid injection (kWh)." Models should be developed based on features such as temperature, energy consumption, atmospheric pressure, and/or wind speed, among other characteristics of a given time point.

Two test datasets are available for validating the model's accuracy:

- energia_202301-202304.csv
- meteo_202301-202304.csv

In these test datasets, the class/target/label referring to the amount of energy injected into the electricity grid is not provided for each record. The task for each group is to predict this class/target/label using the model developed.

An example submission file (dummy_submission.csv) is provided, assuming that the amount of energy injected into the electricity grid is None for all records in the test dataset.

3.2 Features of Energy Datasets

The features of the energy datasets include:

- **Date** - the timestamp associated with the record, representing the day;
- **Time** - the time associated with the record;
- **Normal (kWh)** - amount of electricity consumed, in kWh, from the electricity grid, in a period considered normal in daily bi-hourly cycles (off-peak hours);
- **Economic Hours (kWh)** - the amount of electricity consumed, in kWh, from the electricity grid, in a period considered economic in daily bi-hourly cycles (off-peak hours);
- **Self-consumption (kWh)** - amount of electricity consumed, in kWh, from solar panels;
- **Injection into the grid (kWh)** - amount of electricity injected into the grid, in kWh, from solar panels.

3.3 Weather Dataset Features

The features of the weather dataset include:

- **dt** - the timestamp associated with the record;
- **dt_iso** - the date associated with the record, up to the second;
- **city_name** - the location in question;
- **temp** - temperature in ^{o}C;
- **feels_like** - thermal sensation in ^{o}C;

- **temp_min** - minimum temperature felt in ºC;
- **temp_max** - maximum temperature felt in ºC;
- **pressure** - felt atmospheric pressure in atm;
- **sea_level** - atmospheric pressure felt at sea level in atm;
- **grnd_level** - atmospheric pressure felt at local altitude in atm;
- **humidity** - humidity in percent;
- **wind_speed** - wind speed in meters per second;
- **rain_1h** - average value of precipitation;
- **clouds_all** - level of cloudiness in percent;
- **weather_description** - qualitative assessment of the weather.

3.4 Methodology

3.4.1 Data Preparation

The initial step in our methodology involves consolidating the energy and weather datasets. Two separate energy datasets, namely `energia_202109-202112.csv` and `energia_202201-202212.csv`, were read into DataFrames (`energy1` and `energy2`). Similarly, weather datasets (`meteo_202109-202112.csv` and `meteo_202201-202212.csv`) were read into DataFrames (`weather1` and `weather2`).

To combine the datasets, both energy and weather DataFrames were concatenated along the rows using the `pd.concat` function, resulting in `combined_energy` and `combined_weather` DataFrames. This process ensures a unified dataset for training and tuning Machine Learning models.

3.4.2 Data Cleaning

Certain columns deemed irrelevant for model training and testing were dropped from the combined datasets. For the weather data, the 'dt' (timestamp) and 'city_name' columns were removed, and for the energy data, the 'Data' (date) and 'Hora' (time) columns were excluded. The cleaning process ensures a streamlined dataset focused on relevant features.

3.4.3 Merging Datasets

To create comprehensive datasets for both training and testing, an inner join was performed on the 'dt_iso' column, which represents the timestamp associated with each record. This merge operation combined the energy and weather datasets into `combined_data` for training and `test_data` for testing.

3.4.4 Addition of More opensource API

Data was pulled from: https://github.com/open-meteo/open-meteo

To enhance the feature set for our model, additional data was incorporated. The final dataset now includes the following columns: 'Normal (kWh)', 'Horário Económico (kWh)', 'Autoconsumo (kWh)', 'Injeção na rede (kWh)', 'dt_iso', 'temp', 'feels_like', 'temp_min', 'temp_max', 'pressure', 'sea_level', 'grnd_level', 'humidity', 'wind_speed', 'rain_1h', 'clouds_all', 'weather_description', 'temperature_2m', 'relative_humidity_2m', 'dew_point_2m', 'apparent_temperature', 'precipitation', 'rain', 'snowfall', 'snow_depth', 'weather_code', 'pressure_msl', 'surface_pressure', 'cloud_cover', 'cloud_cover_low', 'cloud_cover_mid', 'cloud_cover_high', 'wind_speed_10m',

'wind_speed_100m', 'wind_direction_10m', 'wind_direction_100m', 'wind_gusts_10m', 'is_day', 'sunshine_duration', 'shortwave_radiation', 'direct_radiation', 'diffuse_radiation', 'direct_normal_irradiance' 'terrestrial_radiation', 'shortwave_radiation_instant', 'direct_radiation_instant', 'diffuse_radiation_instant', 'direct_normal_irradiance_instant', 'terrestrial_radiation_instant'.

3.4.5 Additional Data - holidays

In order to enhance the dataset, a Python function was implemented to generate information regarding weekends and holidays in Braga. This function introduced a new column, 'holidays,' into the existing dataset, thereby enriching the final training dataset.

3.4.6 Data Preprocessing

Columns with a significant number of missing values, such as 'sea_level' and 'grnd_level,' were dropped. For the 'rain_1h' column, missing values were imputed with the average value of 'rain_1h' to address the NaN entries.

Additional encoding was applied to the 'dt_iso' column to create more features, including distinctions between weekends and weekdays, year, month, day, hour, minute, day of the week, and an indicator for weekends (is_weekend).

3.5 Training

XGBoost was employed for the training phase. Our choice of this algorithm was influenced by the findings in the paper titled "When Do Neural Nets Outperform Boosted Trees on Tabular Data?" available at https://arxiv.org/abs/2305.02997.

To ensure that our model was not overfitting, we assessed its performance using cross-validation.

We thoroughly examined and compared classification reports at different stages of training, paying close attention to metrics such as precision, recall, F1-score, and accuracy.

GridSearchCV was utilized to optimize hyperparameters for improved model performance.

Additionally, we experimented with ensemble techniques, incorporating base models such as XGBoost, HistGradientBoosting, and CatBoost.

3.6 Results

The base XGBoost model, without incorporating additional data, achieved an accuracy score of 0.86 on the Kaggle platform. After augmenting the dataset with additional meteorological information, the accuracy improved to 0.88.

Ensemble techniques demonstrated exceptionally high accuracy scores during the training-testing phase, reaching nearly 90 percent. However, validation revealed that these models were prone to overfitting.

The inclusion of holiday data was based on the assumption that energy consumption would be higher on holidays and weekends, assuming people stay at home. This addition resulted in a slight improvement, raising the accuracy to 89 percent however the assumption was wrong and that lead to slightly bad results

3.7 Results Assessment

In conclusion, the findings emphasize that the addition of more relevant data significantly enhances model accuracy. It proves to be the most influential factor in model improvement. However, caution is advised when employing ensemble models, as they have the potential to overfit. Proper validation strategies are crucial when incorporating these techniques.

4 Overall Conclusion

This report encapsulates our journey through both dataset exploration and participation in a Kaggle competition, highlighting the pivotal role of methodology, advanced tools, and hands-on experience in the realm of data science. Engaging with real-world datasets and actively participating in a competition provided us with invaluable practical insights. The challenges encountered not only tested our skills but also fueled our enthusiasm for hands-on work with real-world tools.